(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 November 2002 (07.11.2002)

PCT

(10) International Publication Number
WO 02/089000 A1

(51) International Patent Classification⁷: G06F 17/30, H04L 12/00

(21) International Application Number: PCT/AU02/00518

(22) International Filing Date: 26 April 2002 (26.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PR 4589 26 April 2001 (26.04.2001) AU

(71) Applicant *(for all designated States except US)*: IINET LIMITED [AU/AU]; 263 Adelaide Terrace, Perth, Western Australia 6000 (AU).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: HASELDEN, Mark, David [AU/AU]; 263 Adelaide Terrace, Perth, Western Australia 6000 (AU). BELL, Clayton, Andrew [AU/AU]; 263 Adelaide Terrace, Perth, Western Australia 6000 (AU).

(74) Agent: WRAY & ASSOCIATES; Level 6, 239 Adelaide Terrace, Perth, Western Australia 6000 (AU).

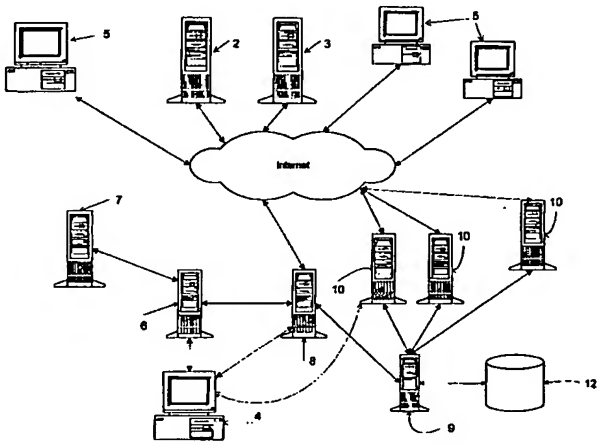
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A SYSTEM FOR CACHING DATA DURING PEER-TO-PEER DATA TRANSFER



(57) Abstract: system for caching data during peer-to-peer data transfer between computers in a network of computers, the system including a database (12) for dynamically storing information regarding the data available for transfer, proxy server (8) and cache manager (9), when combined, operable to transfer data selected from the available data to user (4) in response to a request for the selected data from the user (4); and cache server (10) for storing a copy of data already transferred by the system. The proxy server (8) and cache manager (9), in combination, are further operable to search the database (12) in response to a request for the data selected, to determine if the data selected is stored in the cache server (10), and to transfer the data selected from the cache server (10) to the user (4) if the data selected is stored therein, and, if the data selected is not available in the database (12), to trasnsfer the data selected from a remote user (5) in the network having the data selected stored therein, to the cache server (10)and to the user (4).

WO 02/089000 A1

"A system for caching data during peer-to-peer data transfer"

Field of the Invention

The present invention relates to a system for caching data during peer-to-peer data transfer in a server mediated peer-to-peer file-sharing system, particularly, although not exclusively, using the Internet, and for sharing audio files.

Throughout the specification, unless the context requires otherwise, the word "comprise" or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or group of integers but not the exclusion of any other integer or group of integers.

Background Art

The following discussion of the background to the invention is intended to facilitate an understanding of the present invention. However, it should be appreciated that the discussion is not an acknowledgement or admission that any of the material referred to was published, known or part of the common general knowledge of the relevant person skilled in the art at the priority date of the application.

The Internet is a network of computers that communicate via communication links such as a telephone network, and allow the exchange of information between the computers. The World Wide Web is a part of the Internet, which allows networked computers to send graphical images and data (including audio and video information) between the computers.

Typically, computers – commonly referred to as servers – provide data (or "web pages") in the form of text and graphical images that can be downloaded to remote terminals such as a personal computer ("PC"), where a user can view the web pages. These pages can be used to display information on the display of the remote terminal, for example to give information about a particular service provided by the host of a web site, or products and services provided by the host. The pages may include video and audio information and allow viewers of the web page to view the information. These web pages are viewed at the remote terminal using a so-called web browser.

Each web site (and individual web pages) is uniquely identifiable by means of a web address, or Uniform Resource Locater ("**URL**"). Hyper Text Mark Up Language ("**HTML**") is a text and graphics formatting software that is commonly used to produce and view web pages, and the protocol used by web browsers and servers transferring HTML based files is the Hypertext Transfer Protocol ("**HTTP**"). In data transfer, when one application is in communication with another on a host computer, that application is specified in each data transmission by using its so-called port number. The well-known port number for HTTP transmission is port 80, for example.

The Transmission Control Protocol - Internet Protocol ("**TCP/IP**") is the standard communications protocol for sending data over the Internet. As is well known in the art, protocols that work together as a group are commonly referred to as a protocol stack, with different layers. The HTTP protocol operates at a higher layer than the TCP/IP protocol.

Data is sent in packets (datagrams), which are routed via an Internet Protocol address ("**IP address**"). The required IP address is obtained by cross-referencing the specified URL.

When a user wishes to access the Internet they usually connect to an Internet Service Provider ("**ISP**"). The ISP receives a request from the user to access a specific web site and connects the user's terminal to the required server having the requested URL. The method of connection and subsequent communication is as has been described above.

However, to reduce the amount of information that is transmitted over the Internet, ISPs use a caching system where data that is frequently accessed is stored at the ISP. When that data is next requested by a user, it is provided from the local cache rather than connecting the user's terminal to the required server. Existing caching systems use the known port number of HTTP transfers to capture requests for data and to redirect those requests to a local cache, if appropriate.

The Internet can also be used to facilitate the sharing of files between users. One popular use is the sharing of audio files – typically those in the so-called MP3 format – using a server mediated peer-to-peer file sharing system. In this case, users access a Web site for details of other remote users connected to the web site who are willing to share files, the user then accesses another remote user's computer directly and downloads the required files from that remote users computer. This is illustrated in Figure 1, and is described in more detail below.

In the known file-sharing system 1, a user 4 accesses a file-sharing system meta-server 2 using the Internet and the services of his ISP. Typically, the user 4 will be accessing the meta-server 2 using a terminal such as a personal computer equipped with a modem and the usual user interfaces as is well known. The meta-server 2 will return the address of one or more file-sharing system servers 3 which the user 4 will use to access details of files available for download. Upon connection to the system server 3 – using for example, known login techniques - the user 4 will search or browse for files of interest and available for sharing by other remote users 5 of the system and currently connected to it. This is done by sending either a SEARCH_REQUEST or BROWSE_REQUEST to the system server 3. In a SEARCH_REQUEST the user 4 will specify certain characteristics of a file that is required, and the lists of all remote users 5 connected to the system servers 3 will be examined to see if a requested file, with these characteristics, is available to share. In a BROWSE_REQUEST the user will supply details of a particular known remote user, and all files that the remote user (if connected to the system server 3) is willing to share will be examined. The system server 3 then issues either a SEARCH_RESPONSE or BROWSE_RESPONSE back to the user 4 containing information on the files that are of interest to the user 4. Typically the results of the search or browse operation will be displayed on the user's remote terminal. If the user wishes to download one of the files, he issues a DOWNLOAD_REQUEST to the system server 3. The system server 3 then issues a DOWNLOAD_ACK message that contains the address of the remote user 5 who has the file available for download. The user 4 – without intervention from the system server 3 – makes a connection to the remote user 5 and the file is copied to the user 4.

MP3 audio files are often large in size, and, because this type of file sharing can be extremely popular, it can use a significant proportion of an ISP's bandwidth. Because this file transfer is peer-to-peer, it is not carried over any specific port – the port to be used for a particular file transfer is negotiated between the two communicating peers on a transfer by transfer basis. This makes it difficult to cache. Further, the data is transient – it is only accessible while the remote user 5 that is sharing the file is connected to the Internet, further increasing the difficulty of caching the data.

Disclosure of the Invention

According to a first aspect of the present invention, there is provided a system for caching data during peer-to-peer data transfer between computers in a network of computers, the system including:

a first database for dynamically storing information regarding the data available for transfer from one or more computers in the network of computers;

control means operable to transfer data selected from the available data to a first computer in the network in response to a request for the selected data from the first computer; and

storage means for storing a copy of data already transferred by the system;

wherein the control means is further operable to search the first database in response to a request for the selected data, to determine if the data is stored in the storage means, and to transfer the data from the storage means to the first computer if the data is stored therein, and, if the data is not available in the first database, to transfer the selected data from another computer in the network having the selected data stored therein, to the storage means for storage therein, and to the first computer.

This has the advantage of allowing data that has already been transferred by the system to be easily downloaded to additional users without having to connect to the original storage medium. Thus, files shared through a peer-to-peer file transfer system can be easily cached.

Preferably, the control means comprises a first server and a second server, the first server being coupled to a remote server for receiving information regarding the available data selected therefrom and for storing the information in the first database, the second server including a second database having information to identify the data selected stored in the storage means and their storage locations therein and being coupled to the first server for communication therewith.

More preferably, the second database identifies the data selected stored in the storage means by means of hashing methods or other unique identified encoding methods.

Still more preferably, the second database identifies the data selected stored in the storage means by means of an MD5 hash value.

Preferably, the second server may be operable, in response to a request from the first server, to search the second database for the identifying data to thereby determine if the data selected is stored in the storage means, and its storage therein.

Preferably, the second server may be operable, if the data selected is determined to be stored in the storage means, to send information on the storage location to the first server, the first server being further operable, in response to the received storage location information, to send this information to the first computer.

Preferably, the second server may be operable, if the selected data is determined not to be stored in the storage means, to send information on an available for storage location for the selected data to the first server, the first server being further operable, in response to the received storage location information, to send this information to the first computer.

According to a second aspect of the invention there is a method of caching data during peer-to-peer data transfer including:

selecting data to transfer;

searching a first database of dynamically stored information regarding data available for transfer from one or more computers in a network of computers to determine if the selected data is stored in a storage means;

if the search of the first database determines that the selected data is stored in the storage means, transferring the selected data from the storage means to a first computer in the network of computers; and

if the search of the first database determines that the selected data is not stored in the storage means, transferring the selected data from another computer in the network of computers having the selected data stored therein to the storage means and to the first computer.

Preferably, the method includes the steps of

receiving information regarding the data available for transfer from a remote server; and

selecting data to transfer from the received information.

According to a third aspect of the invention there is provided data transferred between computers in a network of computers, the data having been transferred by a control means in response to a request from a first computer in the network wherein the data has been selected from a set of dynamic data available for transfer stored in a first database and the control means has searched a storage means that stores a copy of data previously transferred by the control means to determine if the data is stored in the storage means and, if the control means determines that the data is stored in the storage means, to initiate the transfer of the data to the first computer from the storage means and, if the control means determines that the data is not stored in the storage means, to initiate the transfer of the data to the first computer and the storage means from a second computer in the network having the selected data stored therein.

Preferably, the control means determines whether the data is stored in the storage means by searching a second database containing information that identifies the data stored in the first database and their storage locations.

Brief Description of the Drawings

One specific embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings, of which:

Figure 1 is a schematic representation of a peer-to-peer file sharing system of the prior art; and

Figure 2 is a schematic representation of a system for caching data during peer-to-peer data transfer of the present invention.

Best Mode(s) for Carrying Out the Invention

Figure 2 illustrates a system for caching data during peer-to-peer data transfer of the present invention. Those features that are known from the prior art, and discussed above in the preamble, will use the same reference numerals for clarity.

In the embodiment, a user 4 is connected to the Internet via his ISP, in a known manner, and can access web sites by typing in the desired URL in his web browser.

The ISP is provided with an ISP server 6 which acts as a gateway to provide connection to the Internet for the user 4 (as is well known in the art). The ISP is also provided with a cache system, which communicates with the ISP server 6. The cache system comprises a proxy server 8, a cache director 7, a cache manager 9 and one or more cache servers 10 – whose functions will be described in more detail below.

When the user 4 wishes to download files from a remote user 5, he starts a file-sharing application that requests connection to a file sharing service using the TCP/IP protocol.

In response to the connection request from the user 4, the ISP server 6 will redirect the user 4 to a cache director 7 for connection thereto. In response to this connection, the cache director 7 returns the address of the proxy server 8 via ISP server 6. The ISP server 6 is able to capture the initial connection request because it uses the TCP/IP protocol with a known port number.

It should be appreciated that in other embodiments, the cache director 7 may be omitted. It is used in this embodiment to emulate the meta-server of existing peer-to-peer file sharing systems such as the Napster system operated by Napster Inc. of Redwood City, California.

The user 4 then connects to the proxy server 8 via ISP server 6. Upon receipt of a connection attempt from the user 4, the proxy server 8 connects to the meta-server 2 by a direct TCP/IP connection and retrieves an IP address of a system server 3. Meta-server 2 contains a large collection of valid system server addresses – the proxy server 8, retrieves one of these by querying meta-server 2 The proxy server 8 then makes a direct connection to the system server 3 using the address retrieved from its query of meta-server 2.

The proxy server 8 acts as an intermediary between the user 4 and the system server 3, and is operable to forward all messages it receives from the user 4 to the system server 3, and vice versa. These messages are forwarded verbatim until the system server 3 sends either a BROWSE_RESPONSE or SEARCH_RESPONSE message, in reply to a BROWSE_REQUEST or SEARCH_REQUEST message from the user 4 to the system server 3 to browse or search the files available for download.

When a BROWSE_RESPONSE or SEARCH_RESPONSE message is received by the proxy server 8, the proxy server 8 is operable to store details of the files available for sharing in its memory. These details include the filename, the MD5 hash of the first 300kB, the file size, encoding bitrate, length (time), the nickname (or nick) of the user sharing the file and the IP address of the user sharing the file. Some or all of these details will also be displayed at the user's remote terminal in the usual manner.

If there is a file that the user 4 wishes to download, then the user 4 issues a DOWNLOAD_REQUEST message that is sent – via proxy server 8 – to system server 3. The system server 3 then responds with a DOWNLOAD_ACK message, which is sent to the proxy server 8.

The proxy server 8, in response to the received DOWNLOAD_ACK message, connects to cache manager 9, which – in response - will return the address of one of the cache servers 10 to the proxy server 8.

The cache manager 9 includes a database 12. In the present embodiment, the database is an SQL database, but could be any mechanism for persistent storage. The cache manager 9 will accept one of four request types – namely, ASK=PROXY; ASK=CACHE, ASK=UPDATECACHED, AND ACTION=DELETECACHED. These will be discussed in more detail below.

The address returned by the cache manager 9 in response to a received message from the proxy server 8 will depend upon whether the file requested by the user 4 is already stored (or cached) locally in one of the cache servers 10, or whether it must be downloaded from a remote user 5 who has the file available for sharing.

The proxy server 8 will send an ASK=PROXY request to the cache manager 9 to request a file with a specific filename, from a specific remote user, with a specific MD5 hash value, in response to a received DOWNLOAD_ACK message from the system server 3. The cache manager 9 will search its database 12, firstly to determine if a file exists with the same MD5 hash value. If this is the case, the cache manager 9 will determine that it has found the requested file and will return the address of the cache server 10 that contains that file. The MD5 hash is an algorithm used to provide a degree of certainty that two files are the same, and is well known to the person skilled in the art.

If the cache manager 9 cannot detect a match for the MD5 hash value, then the cache manager 9 will attempt to find an approximate match. The algorithm for this is to, firstly, strip the path information from the filename, and attempt to match the filename exactly. If a match is found, then the cache manager 9 will then compare the size of the requested file with the size of the stored file and the encoding bit rate of the requested file with that of the stored file. If the encoding bit rates match, and the requested file size is no longer than that of the stored file (to prevent partially downloaded files being stuck in the cache), then a match will be recorded, and, again, the address of the cache server 10 containing the file will be returned. The algorithm is readily extendable to include matching on any of the stored details of the file.

Should there be no exact or approximate match, then an assumption is made that the file is not stored in one of the cache servers 10 i.e. that it is not stored locally. In this case, the cache manager 9 selects a cache server 10 that has available storage capacity and the address of this cache server 10 will be returned – as it will be to this cache 10 that the requested file will be downloaded to. A record is then written into database 12 that contains details passed in the ASK request, including the address of the remote user 5 from which the file should be downloaded.

In response to the received cache server 10 address, the proxy server 8 will rewrite the DOWNLOAD_ACK message with the address of the respective cache server 10 (rather than the address of the remote user 5, which will have been included in the DOWNLOAD_ACK message sent from the system server 3 to the proxy server 8). The proxy server 8 will then send the rewritten DOWNLOAD_ACK message to the user 4.

Upon receipt of the rewritten DOWNLOAD_ACK message, the user 4 will then attempt to directly connect to the particular cache server 10 (rather than the remote client 5), identified by the address inserted in the rewritten DOWNLOAD_ACK message from the proxy server 8.

In response to a connection by the user 4, the cache server 10 issues a ASK=CACHE request to the cache manager 9 denoting a request to the cache manager 9 for the MD5 hash value for the requested file. The cache server 10 will send the username of the remote user 5 and the full pathname for the requested file – as sent with the request from the user 4 to the cache server 10. In response to this request, the cache manager 9 will compare this data with that stored in the database 12, and return the MD5 value. Using this MD5 hash value, the cache server 10 will search its own memory for a file identified by the returned MD5 hash value. If the memory contains this file, then this is copied from the cache server 10 to the user 4 in the usual way. This second lookup is necessary so that the cache server can serve the correct file on request from the user – the protocol used in the Napster system referred to above specifies that download requests contain only the filename to be downloaded, while the cache server 10 only indexes on the MD5 sum. The cache manager 9, via the data stored in the database 12 is the only location where such a correlation can be made.

If the cache server 10 does not find a copy of this file in its database from the returned MD5 hash value, then it assumed that the file is not available locally and must be downloaded from the remote user 5. In this case, the cache server 10 retrieves the IP address of the remote user 5 that is sharing the file from the information stored when the proxy server 8 received the DOWNLOAD_ACK message. The cache server 10 then makes an outgoing connection to the remote user 5, and requests a copy of the file from the remote user 5 in a known manner. The downloaded file is copied to the user 4, and is also copied into the memory on the cache server 10 at the same time.

When the file has been copied to the memory, the cache server 10 will send an ACTION=UPDATECACHED message to the cache manager 9 informing it of the file transfer, and including details of the newly stored file, so that it may be copied directly from the cache manager 10, should another request be made for that particular file. This flags that the file has been downloaded completely and therefore should not be cleaned up from the cache if a script were to be run cleaning the cache information. Its explicit information is to update the database state column in the database 12, for a particular file from 'downloading" to "cached".

If an error arose during download, then the cache manager 9 will also be informed by an ACTION=DELETECACHED request from the cache server 10 to the cache manager 9. In this situation, it is undesirable for further requests to be directed to the partially downloaded file. The DELETECACHED clause directs the cache manager 9 to remove the database row in the database 12 referencing the partially downloaded file. If an error does occur, the user 4 will be advised accordingly and may retry if desired.

All the messages and data transfer discussed above are carried out in accordance with the TCP/IP protocol.

The cache manager 9 and cache servers 10 operate a least recently used policy to removes files from the cache as needed, as is known in the art.

It should be appreciated by the person skilled in the art that the scope of this invention is not limited to the particular embodiment described above. In particular hashing methods other than the MD5 method may be used. Further, and in the alternative, unique identifier encoding methods other than hashing methods may be used.

Claims

1. A system for caching data during peer-to-peer data transfer between computers in a network of computers, the system including:

    a first database for dynamically storing information regarding the data available for transfer from one or more computers in the network of computers;

    control means operable to transfer data selected from the available data to a first computer in the network in response to a request for the selected data from the first computer; and

    storage means for storing a copy of data already transferred by the system;

    wherein the control means is further operable to search the first database in response to a request for the data selected, to determine if the data selected is stored in the storage means, and to transfer the data selected from the storage means to the first computer if the data selected is stored therein, and, if the data selected is not available in the first database, to transfer the data selected from another computer in the network having the data selected stored therein, to the storage means therein, and to the first computer.

2. A system for caching data during peer-to-peer data transfer as claimed in claim 1, wherein the control means comprises a first server and a second server, the first server being coupled to a remote server for receiving information regarding the available data selected therefrom and for storing the information in the first database, the second server including a second database having information to identify the data selected stored in the storage means and their storage locations therein and being coupled to the first server for communication therewith.

3. A system for caching data during peer-to-peer data transfer according to claim 2 wherein the second database identifies the data selected stored in the storage means by means of hashing methods or other unique identified encoding methods.

4. A system for caching data during peer-to-peer data transfer according to claim 3 wherein the second database identifies the data selected stored in the storage means by means of an MD5 hash value.

5. A system for caching data during peer-to-peer data transfer according to any one of claims 2 to 4, wherein the second server may be operable, in response to a request from the first server, to search the second database for the identifying data to thereby determine if the data selected is stored in the storage means, and its storage therein.

6. A system for caching data during peer-to-peer data transfer according to any one of claims 2 to 5, wherein the second server may be operable, if the data selected is determined to be stored in the storage means, to send information on the storage location to the first server, the first server being further operable, in response to the received storage location information, to send this information to the first computer.

7. A system for caching data during peer-to-peer data transfer according to any one of claims 2 to 6, wherein the second server may be operable, if the selected data is determined not to be stored in the storage means, to send information on an available for storage location for the selected data to the first server, the first server being further operable, in response to the received storage location information, to send this information to the first computer.

8. A method of caching data during peer-to-peer data transfer including:

selecting data to transfer;

searching a first database of dynamically stored information regarding data available for transfer from one or more computers in a network of computers to determine if the selected data is stored in a storage means;

if the search of the first database determines that the selected data is stored in the storage means, transferring the selected data from the storage means to a first computer in the network of computers; and if the search of the first database determines that the selected data is not stored in the storage means, transferring the selected data from another computer in the network of computers having the selected data stored therein to the storage means and to the first computer.

9. A method of caching data during peer-to-peer data transfer according to claim 8, further including the steps of

receiving information regarding the data available for transfer from a remote server; and

selecting data to transfer from the received information.

10. Data transferred between computers in a network of computers, the data having been transferred by a control means in response to a request from a first computer in the network wherein the data has been selected from a set of dynamic data available for transfer stored in a first database and the control means has searched a storage means that stores a copy of data previously transferred by the control means to determine if the data is stored in the storage means and, if the control means determines that the data is stored in the storage means, to initiate the transfer of the data to the first computer from the storage means and, if the control means determines that the data is not stored in the storage means, to initiate the transfer of the data to the first computer and the storage means from a second computer in the network having the selected data stored therein.

11. Data according to claim 10 wherein the control means determines whether the data is stored in the storage means by searching a second database containing information that identifies the data stored in the first database and their storage locations.

12. A system for caching data during peer-to-peer data transfer substantially as described herein with reference to Figure 2.

13. A method of caching data during peer-to-peer data transfer substantially as described herein with reference to Figure 2.

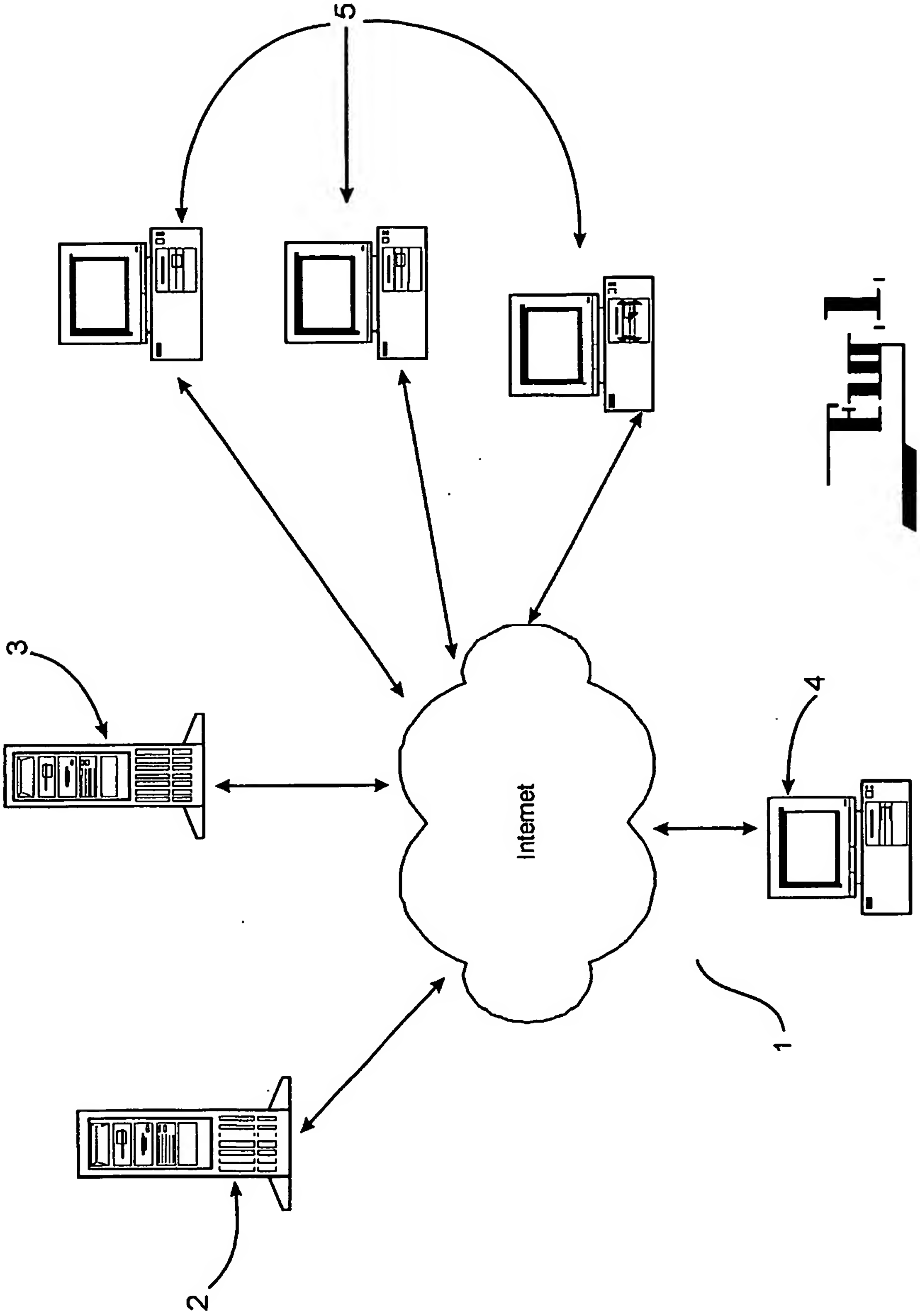


Fig 1

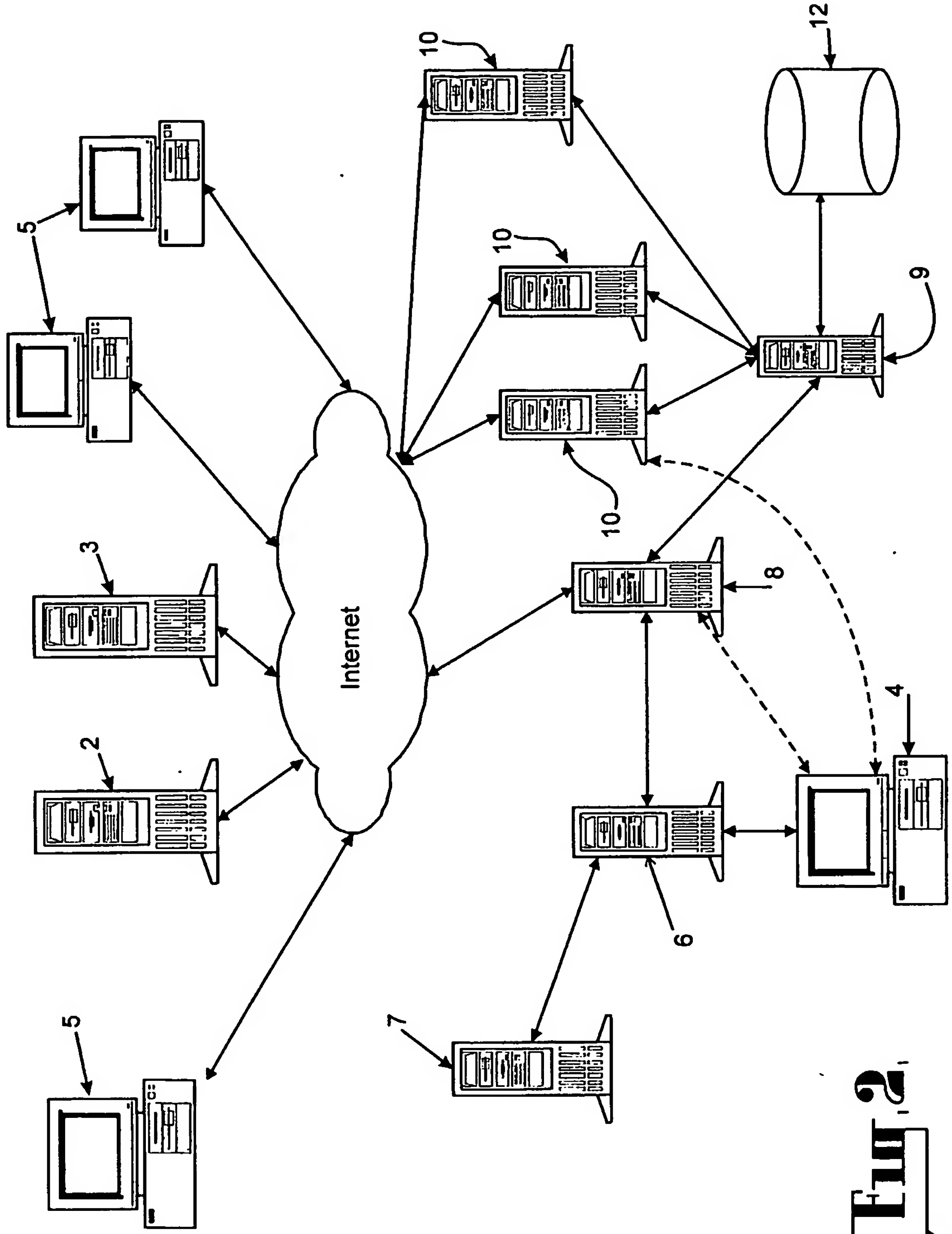


Fig 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU02/00518

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. ^{7}: G06F 17/30, H04L 12/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPAT, USPTO, (cache, database, peer-to-peer)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6167438 A (Yates et al) 26 December 2000 Whole Document	1-13
X	US 6092080 A (Gustman) 18 July 2000 Whole Document	1-13
X	WO 00/42519 A1 (Edgix Corporation) 20 July 2000 Whole Document	1-13

□ Further documents are listed in the continuation of Box C ☒ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
12 July 2002

Date of mailing of the international search report
25 JUL 2002

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustralia.gov.au
Facsimile No. (02) 6285 3929

Authorized officer

R.H. STOPFORD
Telephone No : (02) 6283 2177

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/AU02/00518

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
US	6167438	AU	7388498	EP	0983558	WO	9853410
		NZ	501380	ZA	9804298		
US	6092080		NONE				
WO	00/42519	AU	2847700				

END OF ANNEX